

Sparse solution of the Lyapunov equation for large-scale interconnected systems[★]

Aleksandar Haber and Michel Verhaegen

Delft University of Technology, 2628 CD Delft, The Netherlands

Abstract

We consider the problem of computing an approximate banded solution of the continuous-time Lyapunov equation $\underline{A}\underline{X} + \underline{X}\underline{A}^T = \underline{P}$, where the coefficient matrices \underline{A} and \underline{P} are large, symmetric banded matrices. The (sparsity) pattern of \underline{A} describes the interconnection structure of a large-scale interconnected system. Recently, it has been shown that the entries of the solution \underline{X} are spatially localized or decaying away from a banded pattern. We show that the decay of the entries of \underline{X} is faster if the condition number of \underline{A} is smaller. By exploiting the decay of entries of \underline{X} , we develop two computationally efficient methods for approximating \underline{X} by a banded matrix. For a well-conditioned and sparse banded \underline{A} , the computational and memory complexities of the methods scale linearly with the state dimension. We perform extensive numerical experiments that confirm this, and that demonstrate the effectiveness of the developed methods. The methods proposed in this paper can be generalized to (sparsity) patterns of \underline{A} and \underline{P} that are more general than banded matrices. The results of this paper open the possibility for developing computationally efficient methods for approximating the solution of the large-scale Riccati equation by a sparse matrix.

1 Introduction

Large-scale interconnected systems consist of the interconnection of a large number of dynamical subsystems [1–15]. The focus of this paper is on the large-scale interconnected systems described by state-space models with (sparse) banded matrices¹. The importance of this class of interconnected systems is best illustrated by the fact that state-space models with banded matrices are obtained by discretizing Partial Differential Equations (PDEs) using the finite difference or finite element methods [16, 17]. Each discretization node or a group of discretization nodes can be seen as a subsystem, and the discretization mesh can be interpreted as a network of dynamical systems [17, 18].

The Lyapunov equation is ubiquitous in systems and control theory and in signal processing [19]. For us, the most interesting application of the Lyapunov equation, is in the methods for solving the optimal control (estimation) problems of large-scale systems [20, 21]. Namely, the solution of the Linear Quadratic (LQ) optimal control problem can be found by solving the Riccati equation. A widely used method for solving the Riccati equation is the Newton method [20, 22–27]. In each step of the Newton method it is necessary to solve the Lyapunov or

Sylvester equations (or the Stein equation for discrete-time systems, see for example [28]). Further applications of the Lyapunov and Sylvester equations can be found in [19, 21].

The goal of this paper is to analyze the solution of the Lyapunov equation for large-scale interconnected systems and to develop computationally efficient methods for approximating the solution by a (sparse) banded matrix. Specifically, we consider the large-scale, continuous-time Lyapunov equation:

$$\underline{A}\underline{X} + \underline{X}\underline{A}^T = \underline{P} \quad (1)$$

where $\underline{A} \in \mathbb{R}^{Nn \times Nn}$ is a (symmetric) negative definite, banded matrix describing the global dynamics of an interconnected system, $\underline{X} \in \mathbb{R}^{Nn \times Nn}$ is a solution that we are searching for, $\underline{P} \in \mathbb{R}^{Nn \times Nn}$ is a banded, negative definite matrix, N is a large number representing the total number of subsystems and $n \ll N$ is the state order of subsystems. Obviously, finding the solution of (1) is a computationally challenging task. The computational challenges in solving the Lyapunov equation for large-scale systems are perhaps best described by the author of the recent survey [21] on linear matrix equations: “A distinctive feature in the large-scale setting is that coefficient matrices (the matrices \underline{A} and \underline{P}) may be sparse, the solution matrix is usually dense and thus impossible to store in memory... For \underline{A} in the order of 10^4 or larger the solution cannot be stored explicitly...”

A large variety of methods for solving the large-scale

[★] Corresponding author: A. Haber, *email address*: aleksandar.haber@gmail.com.

¹ The results of this paper can be extended to matrices with more general sparsity patterns.

Lyapunov equation are coping with this problem by searching for a low rank approximation $\tilde{X} = ZZ^T$ to the "true" solution X , where Z is the "tall" matrix that is computed and stored [21]. However, the approximate solution in the form of $\tilde{X} = ZZ^T$ is a completely dense (fully populated) matrix. Taking into account that the Newton method solves the Riccati equation by solving series of the Lyapunov equations, the solution to the Riccati equation is also dense. This implies that the feedback matrix of the LQ control law is also dense. However, for the distributed control of large-scale interconnected systems, we would like to compute a sparse feedback matrix [10, 29, 30]. Namely, a sparse feedback matrix enables us to implement the controller on a network of sensors, actuators and computing units that communicate locally. On the other hand, due to the fact that the computational and memory complexities of multiplying a vector with a sparse matrix are linear, a sparse feedback matrix implies that the centralized LQ control law can be implemented with linear complexity. If it were possible to accurately approximate the solution of the Lyapunov equation by a sparse matrix, then by using the inexact Newton methods [23] it would be possible to determine a sparse approximate solution of the Riccati equation. That is, if the solution of the Lyapunov equation can be accurately approximated by a sparse matrix, then the LQ feedback matrix can also be approximated by a sparse matrix.

In [29–31], several methods have been developed for computing sparse, optimal feedback gains. However, the computational and memory complexities of these methods are $O(N^3)$ and $O(N^2)$, respectively, and consequently, these methods are not applicable to large-scale systems.

The above explained problems motivate us to search for the answers to the following questions:

(1) Is the solution X *spatially localized*? Under the term of a "spatially localized matrix", we understand a matrix whose entries decay quickly in magnitude outside a (sparsity) pattern (for example, outside a bandwidth of a matrix) [32]. Off-diagonally decaying matrices [32–34] are typical examples of spatially localized matrices. Roughly speaking, the off-diagonal elements are bounded by an exponential function that decays away from the main diagonal.

(2) Can the solution of (1) be accurately approximated by a banded matrix, possibly sparse², and can this approximate solution be computed with $O(N)$ computational and memory complexities? There is a strong correlation between this question and the first one, because if a matrix is spatially localized then it can be accurately approximated by a sparse matrix [1].

² Under the term of a "sparse banded matrix", we understand a banded matrix whose bandwidth is much smaller than its dimensions.

Regarding the first question, it has already been shown that X is a spatially localized matrix. Namely, the Lyapunov equation can be rewritten as a linear system of equations, in which the coefficient matrix is a Kronecker sum of the matrix A and the identity matrix. In [35, 36], it has been shown that the inverses of symmetric, positive definite, banded matrices with a Kronecker sum structure are off-diagonally decaying matrices (with a non-monotonic decay) and several bounds on the off-diagonal decay rate have been derived. These important results imply that X is a spatially localized matrix [37]. However, because the upper-bounds on the off-diagonal decay rate presented in [35–37] have a relatively complex integral form, they need to be evaluated numerically. Consequently, from these bounds it might be hard to draw important conclusions on how the condition number of banded A influences the decay rate of entries of X . It should come as no surprise that the condition number of A dominantly determines the decay rate of X , because it is well-known that the decay rates of matrix functions, such as A^{-1} , are primarily determined by the condition number of A [34]. On the other hand, in [32, 35], it has been shown that the matrix exponential of A , denoted by $\exp(tA)$, where t is time, is also an off-diagonally decaying matrix.

By exploiting the fact that the solution of the Lyapunov equation X and the matrix exponential $\exp(tA)$ are spatially localized matrices, we develop two computationally efficient methods for approximating X by a banded matrix. Furthermore, we show that the decay of entries of X is faster if the condition number of A is smaller. Our results indicate that for a well-conditioned and sparse banded A , the proposed methods are able to approximate the matrix X with $O(N)$ complexity. The results of this paper open the possibility for developing computationally efficient methods for approximating the solution of the large-scale Riccati equation by a sparse matrix. Furthermore, the results of this paper can also be generalized to diagonalizable banded matrices A , and to matrices with more general (sparsity) patterns, see Section 3.2.

This paper is organized as follows. In Section 2, we present the problem formulation. In Section 3, we analyze the decay of entries of X and we present a method for estimating the a priori pattern of an approximate solution. In Section 4, we develop two methods for approximating X . In Section 5, we present numerical experiments, and in Section 6 we present conclusions.

2 Problem formulation

2.1 Notation

The notation $X = [x_{i,j}]$ denotes a matrix whose (i, j) entry is $x_{i,j}$, whereas $\tilde{X} = [\tilde{X}_{i,j}]$ denotes a block matrix whose (i, j) entry is the matrix $\tilde{X}_{i,j}$. The notation

$\mathbf{z} = \text{col}(\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_M)$ stands for $\mathbf{z} = [\mathbf{z}_1^T \ \mathbf{z}_2^T \ \dots \ \mathbf{z}_M^T]^T$. An $N \times N$ matrix $X = [x_{i,j}]$ is called a *banded matrix* if there exists an even positive integer s , such that $x_{i,j} = 0$ when $|i - j| > s/2$ [34]. The number s is called the *bandwidth* of X and we say that the matrix X is *s-banded* [34]. For example, a tridiagonal matrix is 2-banded. If $s \ll N$, then X is called a *sparse banded matrix*. If the matrix X_1 has the bandwidth s_1 and the matrix X_2 has the bandwidth s_2 , then the product $X_1 X_2$ has the bandwidth equal to $s_1 + s_2$. The notations $\|X\|_2$ and $\|X\|_F$ denote the 2-norm and the Frobenius norm of X , respectively. The symbol \otimes denotes the Kronecker product, and the operator $\text{vec}(X)$ is a standard "vec" operator [38]. The (column) vector \mathbf{q}_j denotes a vector that has all zeros except 1 on the position j [37]. For example, a matrix Z , having only one non-zero element $z_{i,j}$ at position (i, j) , can be represented by $Z = \mathbf{q}_i z_{i,j} \mathbf{q}_j^T$.

We consider a subsystem \mathcal{S}_i :

$$\mathcal{S}_i \begin{cases} \dot{\mathbf{w}}_i(t) = A_{i,i} \mathbf{w}_i(t) + \sum_{j=i-b, j \neq i}^{i+b} A_{i,j} \mathbf{w}_j(t) \\ \mathbf{y}_i(t) = C_i \mathbf{w}_i(t) \end{cases} \quad (2)$$

where $\mathbf{w}_i(t) \in \mathbb{R}^n$ is the *local state* of the subsystem \mathcal{S}_i and $\mathbf{y}_i(t) \in \mathbb{R}^r$ is the *local output*, $A_{i,j} \in \mathbb{R}^{n \times n}$ and $C_i \in \mathbb{R}^{r \times n}$ and t is time. The state-space model of the *global system* \mathcal{S} is:

$$\mathcal{S} \begin{cases} \dot{\mathbf{w}}(t) = \underline{A} \mathbf{w}(t) \\ \mathbf{y}(t) = \underline{C} \mathbf{w}(t) \end{cases} \quad (3)$$

where $\mathbf{w}(t) = \text{col}(\mathbf{w}_1(t), \dots, \mathbf{w}_N(t))$, $\mathbf{y}(t) = \text{col}(\mathbf{y}_1(t), \dots, \mathbf{y}_N(t))$, $\underline{A} \in \mathbb{R}^{Nn \times Nn}$ and $\underline{C} \in \mathbb{R}^{Nr \times Nn}$. The vectors $\mathbf{w}(t) \in \mathbb{R}^{Nn}$ and $\mathbf{y}(t) \in \mathbb{R}^{Nr}$ are called the *global state* and *global output*, respectively. We assume that the total number of subsystems N is a large number and that $n \ll N$. Furthermore, we assume that $b \ll N$. That is, we assume that the matrix \underline{A} is a sparse banded matrix. The bandwidth of \underline{A} is denoted by $m \ll N$. We assume that the matrix \underline{A} is symmetric and asymptotically stable (although all the methods in this paper can be generalized to banded diagonalizable matrices and to matrices with more general patterns, see Section 3.2). Finally, we assume that the matrix \underline{P} in (1) is a sparse, banded matrix with the bandwidth equal to l , where $l \ll N$. For example, the model (3) can be obtained by discretizing the 2D or 3D partial differential equations using the finite difference methods [16, 17].

Methods for solving (1) will be built upon on the following two representations of the solution \underline{X} [37, 39]. Taking into an account the symmetry of \underline{A} , the (unique) solution of (1) has the integral representation [39]:

$$\underline{X} = - \int_0^\infty \exp(t\underline{A}) \underline{P} \exp(t\underline{A}) dt \quad (4)$$

By vectorizing (1), we obtain:

$$\mathcal{A} \bar{\mathbf{x}} = \bar{\mathbf{p}}, \quad (5)$$

$$\bar{\mathbf{x}} = \mathcal{A}^{-1} \bar{\mathbf{p}} \quad (6)$$

where $\mathcal{A} \in \mathbb{R}^{(Nn)^2 \times (Nn)^2}$, $\bar{\mathbf{x}}, \bar{\mathbf{p}} \in \mathbb{R}^{(Nn)^2}$ are defined by:

$$\mathcal{A} = I \otimes \underline{A} + \underline{A} \otimes I, \quad \bar{\mathbf{x}} = \text{vec}(\underline{X}), \quad \bar{\mathbf{p}} = \text{vec}(\underline{P}) \quad (7)$$

and where I is an $Nn \times Nn$ identity matrix. Based on the fact that the entries of the solution \underline{X} are decaying away from a banded pattern (in some cases they exhibit a damped oscillatory behavior) [35–37], in the sequel we will develop two methods for approximating \underline{X} by a (sparse) banded matrix. The first method looks for an approximate solution by solving a least-squares problem formed on the basis of (5). The least-squares problem is formed by eliminating the columns of \mathcal{A} and the elements of $\bar{\mathbf{x}}$ corresponding to the small elements of \underline{X} that are predicted by the decaying property. The second method consists of the following two steps. In the first step, the integral in (4) is approximated by a banded matrix. This approximation is obtained by exploiting the fact that $\exp(t\underline{A})$ is an off-diagonally decaying matrix [32, 35]. In the second step, the accuracy of this approximate solution is additionally improved by using the gradient projection method [40]. We start with the analysis of the decaying behavior of entries of \underline{X} .

3 Decay rate analysis and a priori pattern of \underline{X}

The goal of this section is to analyze the influence of the condition number of \underline{A} on the decay of entries of \underline{X} , and to develop a method for the prediction of the *a priori (sparsity) pattern* of the approximate solution. Important insights obtained in this section will be used in Section 4 to develop computationally efficient approximation methods.

Definition 1 [32, 34, 36, 41] *We say that an $Nn \times Nn$ matrix $Z = [z_{i,j}]$ is an off-diagonally decaying matrix if there exist $\tau > 0$ and $\rho \in (0, 1)$ such that $|z_{i,j}| \leq \tau \rho^{|i-j|}$ for all $i, j = 1, \dots, Nn$. \square*

The constant ρ is referred to as the *decay rate* of Z [41]. We will use theoretical results developed in [32, 34–36, 42] to analyze the decay rate of \underline{X} , as well as to provide insights on how the condition number of \underline{A} influences the decay rate.

3.1 The influence of the condition number of \underline{A} on decay of \underline{X}

The constants a and b are defined by $a = \lambda_{\min}(\underline{A})$ and $b = \lambda_{\max}(\underline{A})$, where $\lambda_{\min}(\cdot)$ and $\lambda_{\max}(\cdot)$ denote minimal and maximal eigenvalues, respectively. Given that

the matrix \underline{A} is a symmetric, asymptotically stable matrix, we have that $a < 0$ and $b < 0$ and $|b| \leq |a|$. The complexity analysis of computing a and b is discussed in Section 4.3. The condition number of \underline{A} will be denoted by κ . It follows that $\kappa = a/b$.

For presentation clarity, we will first consider a special case, when the matrix \underline{P} is a diagonal matrix, and subsequently we will consider a more general case when the matrix \underline{P} is a banded matrix (or even fully populated). Let us suppose that $\underline{P} = \gamma I$, $\gamma < 0$. Then, because \underline{A} is symmetric, it can be verified that $\underline{X} = ((2/\gamma)\underline{A})^{-1}$ satisfies (1). Due to the fact that $\gamma < 0$, we have $\underline{X} = ((2/\gamma)\underline{A})^{-1} = (|\gamma|/2)(-\underline{A})^{-1}$. Because by assumption \underline{A} is asymptotically stable, symmetric matrix, the matrix $-\underline{A}$ is positive definite, and consequently, from Theorem 2.4 in [34], it follows that $(-\underline{A})^{-1}$ is an off-diagonally decaying matrix. This further implies that $X = [x_{i,j}]$ is an off-diagonally decaying matrix, that is, $|x_{i,j}| \leq \tau \rho^{|i-j|}$, where

$$\tau = \frac{|\gamma|}{2} K_1, \quad K_1 = \frac{1}{|b|} \max \left\{ 1, \frac{(1 + \sqrt{\kappa})^2}{2\kappa} \right\},$$

$$\rho = \left(\frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} \right)^{\frac{2}{m}} \quad (8)$$

From (8) we see that the decay rate of \underline{X} depends on the condition number of \underline{A} . Specifically, if \underline{A} is well-conditioned (κ is close to 1), then the decay rate³ is fast (the number ρ is small). Now, does \underline{X} exhibit a similar behavior when \underline{P} is a sparse banded matrix? Not surprisingly, the answer is yes. Before we show this, it should be first observed that the matrix \mathcal{A} is also an off-diagonally decaying matrix. Namely, from [39] (Theorem 13.16) it follows that:

$$\lambda_{\max}(\mathcal{A}) = 2\lambda_{\max}(\underline{A}) = 2b, \quad \lambda_{\min}(\mathcal{A}) = 2\lambda_{\min}(\underline{A}) = 2a \quad (9)$$

This implies that the condition numbers of \mathcal{A} and \underline{A} are equal. On the other hand, because \underline{A} is m -banded, the matrix \mathcal{A} is m_1 -banded, with $m_1 = Nnm$ [36]. Furthermore, the matrix \mathcal{A} is negative definite. Similarly to the analysis of the decay rate of \underline{X} (for the case of a diagonal \underline{P}), applying the results of Theorem 2.4 in [34] to $-\mathcal{A}$, we conclude that \mathcal{A}^{-1} is an off-diagonally decaying matrix, with the decay rate specified by:

$$\tau_1 = \frac{1}{2|b|} \max \left\{ 1, \frac{(1 + \sqrt{\kappa})^2}{2\kappa} \right\}, \quad \rho_1 = \left(\frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} \right)^{\frac{2}{m_1}} \quad (10)$$

³ Beside κ , we see that the minimal singular value $|b|$ of \underline{A} determines the off-diagonally decaying behavior. Throughout the paper we will assume that $|b|$ is not very small, which implies that the decaying behavior of \underline{A} is primarily determined by κ .

By comparing (8) and (10), we conclude that the decay rates ρ of \underline{X} (for the case of diagonal \underline{P}) and ρ_1 of \mathcal{A}^{-1} , have the same exponential base (determined by κ). Furthermore, we see that because m_1 is by several orders of magnitude larger than m , the decay rate of \mathcal{A}^{-1} is slower than the decay rate of \underline{X} . However, the matrix \mathcal{A}^{-1} is much larger than \underline{X} . Furthermore, taking into account the sizes and the bandwidths of the corresponding matrices, the entries of both matrices that are far away from the corresponding main diagonals or bandwidths, are small when the matrix \underline{A} is well-conditioned. To further analyze the decay rate of \underline{X} , we represent the matrix $\underline{P} = [p_{i,j}]$ as follows [37]:

$$\underline{P} = \sum_{i=1}^{Nn} \sum_{j=1}^{Nn} \underline{P}_{i,j} \quad (11)$$

where $\underline{P}_{i,j} = \mathbf{q}_i p_{i,j} \mathbf{q}_j^T$ and where \mathbf{q}_i and \mathbf{q}_j are defined in Section 2.1. For notation simplicity and presentation clarity, in (11) and throughout the rest of this section, we have formally ignored the fact that most of the entries of \underline{P} are zero (the matrix \underline{P} is l -banded, so its entries $p_{i,j}$ for which $|i - j| > l/2$, are equal to zero). While interpreting the results, it should be kept in mind that the summation in (11) and in subsequent expressions, should be performed only with respect to the indices (i, j) belonging to the bandwidth region. Let $\underline{X}_{i,j} \in \mathbb{R}^{Nn \times Nn}$ be the solution of the Lyapunov equation:

$$\underline{A} \underline{X}_{i,j} + \underline{X}_{i,j} \underline{A} = \underline{P}_{i,j} = \mathbf{q}_i p_{i,j} \mathbf{q}_j^T \quad (12)$$

where $i, j = 1, \dots, Nn$. Then, because of the linearity of the Lyapunov equation it follows that its solution can be decomposed as follows [37]:

$$\underline{X} = \sum_{i=1}^{Nn} \sum_{j=1}^{Nn} \underline{X}_{i,j} \quad (13)$$

where each of $\underline{X}_{i,j}$ is the solution of (12). The decomposition (13) enables us to prove the following theorem.

Theorem 2 Let $\bar{\mathbf{x}}_{i,j} = \text{vec}(\underline{X}_{i,j})$ and let the element of $\bar{\mathbf{x}}_{i,j}$ on the position s , $s = 1, \dots, (Nn)^2$, be denoted by $\bar{x}_s^{i,j}$, then

$$|\bar{x}_s^{i,j}| \leq |p_{i,j}| \tau_1 \rho_1^{|\phi(i,j) - s|} \quad (14)$$

where $\phi(i, j) = (j - 1)Nn + i$ and ρ_1 and τ_1 are defined in (10). Moreover, let the element of $\bar{\mathbf{x}} = \text{vec}(\underline{X})$ on the position s be denoted by \bar{x}_s , then

$$|\bar{x}_s| \leq \tau_1 \sum_{i=1}^{Nn} \sum_{j=1}^{Nn} |p_{i,j}| \rho_1^{|\phi(i,j) - s|} \quad (15)$$

Proof: After applying $\text{vec}(\cdot)$ operator to (12), we obtain:

$$\mathcal{A}\bar{\mathbf{x}}_{i,j} = p_{i,j}\mathbf{q}_{(j-1)Nn+i}, \quad \bar{\mathbf{x}}_{i,j} = p_{i,j}\mathcal{A}^{-1}\mathbf{q}_{(j-1)Nn+i} \quad (16)$$

and according to our notation the vector $\mathbf{q}_{(j-1)Nn+i}$ has 1 on the position $(j-1)Nn+i$. Let the $\phi(i,j) = ((j-1)Nn+i)$ -th column of \mathcal{A}^{-1} be denoted by $\boldsymbol{\theta}_{(i,j)} \in \mathbb{R}^{(Nn)^2}$. Let the element of $\boldsymbol{\theta}_{(i,j)}$ on the position s be denoted by $\theta_s^{(i,j)}$. From (16) it follows that $\bar{\mathbf{x}}_{i,j}$ is equal to the $\phi(i,j)$ -th column of \mathcal{A}^{-1} multiplied by a constant $p_{i,j}$:

$$\bar{\mathbf{x}}_{i,j} = p_{i,j}\boldsymbol{\theta}_{(i,j)} \quad (17)$$

Given that \mathcal{A}^{-1} is an off-diagonally decaying matrix, the absolute values of elements of its each column are bounded by an exponential function that decays away from the element on the main diagonal. The element of $\boldsymbol{\theta}_{(i,j)}$ that is on the main diagonal of \mathcal{A}^{-1} is the element $\theta_r^{(i,j)}$ for which $r = \phi(i,j)$. All this implies that for the entries on the $\phi(i,j)$ -th column of \mathcal{A}^{-1} we can write:

$$|\theta_s^{(i,j)}| \leq \tau_1 \rho_1^{|\phi(i,j)-s|} \quad (18)$$

From (17) and (18) we obtain (14). By vectorizing (13) and using (14), we can similarly prove (15). \square

Although conservative compared to the bounds in [35–37], the bounds in (14) and (15) can be used to analyze the dependence of the decay rate of $\underline{\mathbf{X}}$ on the condition number of $\underline{\mathbf{A}}$. Namely, the upper bound (14) on the entries $\bar{x}_s^{i,j}$ of the vector $\bar{\mathbf{x}}_{i,j}$, has a maximum value for the entry $s = \phi(i,j)$. When, on the other hand, $\bar{\mathbf{x}}_{i,j}$ is transformed back to a matrix format, this maximum corresponds to the (i,j) entry of $\underline{\mathbf{X}}_{i,j}$. Roughly speaking, we also see that the entries of $\underline{\mathbf{X}}_{i,j}$ that are further away from the entry (i,j) , are bounded by a function that decays as the distance between these entries and the entry (i,j) is increased. The decay rate is determined by the condition number of \mathcal{A} (or equivalently by the condition number of $\underline{\mathbf{A}}$). If $\underline{\mathbf{A}}$ is well-conditioned, this decay is fast. Now, taking into account that $\underline{\mathbf{P}}$ is sparse and banded, we have that the upper bounds have maximal values for the entries (i,j) located in the bandwidth region of $\underline{\mathbf{P}}$. On the other hand, from (13), we see that the decaying behavior of $\underline{\mathbf{X}}$ is determined by the sum of locally decaying behaviors of each individual $\underline{\mathbf{X}}_{i,j}$ (it should be remembered that the sum in (13) is only performed over the indices (i,j) for which $p_{i,j}$ is not zero, that is, for the entries inside the bandwidth of $\underline{\mathbf{P}}$). This leads us to the conclusion that the elements of $\underline{\mathbf{X}}$ that are far away from the bandwidth of $\underline{\mathbf{P}}$ should be relatively small for well conditioned $\underline{\mathbf{A}}$. This insights and the insights provided in Section 3.2, will enable us to develop computationally efficient algorithms for approximating $\underline{\mathbf{X}}$ in Section 4.

3.2 Predicting the sparsity pattern of $\underline{\mathbf{X}}$

In Section 4, we will develop computationally efficient methods for approximating $\underline{\mathbf{X}}$. To develop these methods, we need to choose an *a priori* (sparsity) pattern of $\underline{\mathbf{X}}$. From the previous discussion, we may conclude that if the matrix $\underline{\mathbf{A}}$ is well-conditioned then the entries of $\underline{\mathbf{X}}$ that are far away from the bandwidth region of $\underline{\mathbf{P}}$ are small. Consequently, the *a priori* pattern can be chosen as a (sparse) banded matrix. By neglecting the entries of $\bar{\mathbf{x}}$ that are outside this pattern, we can reduce the dimension of the linear system (5), and we can obtain its solution by solving a least squares problem. Furthermore, the reduced coefficient matrix of this system is sparse, and this sparsity can be exploited to quickly compute the solution.

However, a banded *a priori* pattern can sometimes be suboptimal in the sense that the "true" solution $\underline{\mathbf{X}}$ has a large number of small entries inside of the banded *a priori* pattern (this can happen for example when the entries of $\underline{\mathbf{X}}$ exhibit an oscillatory behavior). This can lead to unnecessary increase of the computational and memory complexities of the proposed algorithms. Is there a more optimal *a priori* pattern, that can more accurately capture the "true" behavior of the entries of $\underline{\mathbf{X}}$? Moreover, we might ask ourselves what should be an *a priori* pattern in the case of more general patterns of the coefficient matrices $\underline{\mathbf{A}}$ and $\underline{\mathbf{P}}$?

Due to the Kronecker sum structure of \mathcal{A} , the entries of \mathcal{A}^{-1} exhibit a dominantly oscillatory behavior [35–37]. Relatively good estimates of this behavior are presented in [35–37]. Moreover, from [35–37] it follows that for problems for which $\underline{\mathbf{A}}$ is a Kronecker sum, $\underline{\mathbf{A}} = \mathbf{I} \otimes \mathbf{Z}_1 + \mathbf{Z}_1 \otimes \mathbf{I}$, where \mathbf{Z}_1 is a sparse banded matrix, the entries of $\underline{\mathbf{X}}$ also exhibit an oscillatory behavior. These important insights can be used to design *a priori* patterns that optimally (meaning that the number of non-zero elements is smaller than the number of non-zero elements of a banded matrix) capture the "true" behavior of the entries of $\underline{\mathbf{X}}$. For example, these results can indicate that the *a priori* pattern of $\underline{\mathbf{X}}$ should be a multi-banded matrix (matrix that has series of zero and non zero diagonals below or above the main diagonal, examples are shown in Fig. 9(b) and Fig. 10(d)). On the other hand, decay bounds available in the literature indicate that the results of this paper can be generalized to non-symmetric matrices $\underline{\mathbf{A}}$ and \mathcal{A} . Namely, Theorem 3.5 in [32] proves that functions of diagonalizable banded matrices also exhibit a form of the off-diagonal decay. Moreover, from Theorem 3.4 in [32], it follows that functions of more general class of diagonalizable sparse matrices (not necessarily banded) are spatially localized. The practical potential of the results presented in [35–37] for determining the optimal *a priori* pattern of $\underline{\mathbf{X}}$ will be investigated in our future work.

In this paper, we will present a relatively simple ap-

proach that can give us additional insights into a more optimal a priori pattern of \underline{X} . Furthermore, this approach can be applied to a more general class of sparse matrices \underline{A} and \underline{P} , such as sparse multi-banded matrices or even non-symmetric matrices. Namely, using the Neumann representation of the matrix inverse or the fact that a matrix satisfies its characteristic polynomial, in [43] it has been shown that a relatively good guess of the a priori pattern of \mathcal{A}^{-1} is given by the pattern of the following matrix:

$$\mathcal{C} = I + \mathcal{A} + \mathcal{A}^2 + \dots + \mathcal{A}^{z_1} \quad (19)$$

For simplicity and without the loss of generality, in (19) we will assume that zero entries of \mathcal{C} are not created by incidental cancellations of non-zero entries of matrices produced by taking and summing up powers of \mathcal{A} [43]. If \mathcal{A} is m_1 -banded, then from (19) it follows that \mathcal{C} is $z_1 m_1$ -banded. Next, if the matrix \mathcal{A} is (sparse) banded, then the matrix \mathcal{C} will also be banded, for small values of z_1 . Also, under the same conditions, multi-banded structure of \underline{A} or consequently of \mathcal{A} , will be preserved in \mathcal{C} . The fundamental question that needs to be asked is how large z_1 should be such that the majority of the significant entries of \mathcal{A}^{-1} are captured by the sparsity pattern of \mathcal{C} , and is there a connection between z_1 and the condition number of \mathcal{A} ? The answers to these questions can be obtained by analyzing the accuracy of the expansions from which the expression (19) originates. However, this might be a nontrivial problem. Instead, we use an alternative way to show that for well-conditioned \underline{A} , a relatively good guess of the a priori pattern of \mathcal{A}^{-1} is formed by summing up a relatively low powers of \mathcal{A} in (19). This will be shown by considering the Newton-Schultz iteration [44] for approximating⁴ \mathcal{A}^{-1} . The Newton-Schultz iteration is defined by [44]:

$$\mathcal{B}_{k+1} = \mathcal{B}_k (2I - \mathcal{A}\mathcal{B}_k), \quad k = 0, 1, 2, \dots \quad (20)$$

where \mathcal{B}_k is the approximation of \mathcal{A}^{-1} at the k -th iteration. The iteration is initialized by \mathcal{B}_0 :

$$\mathcal{B}_0 = \frac{2}{a_1^2 + b_1^2} \mathcal{A}^T \quad (21)$$

where a_1 and b_1 are the maximal and minimal singular values of \mathcal{A} , respectively. The accuracy at the k -th iteration of the Newton-Schultz iteration is measured by the norm of $\mathcal{E}_k = I - \mathcal{A}\mathcal{B}_k$. Using the fact that $a_1 = 2|a|$ and $b_1 = 2|b|$ (see (9)), and using the results of [44], it can be

⁴ The Newton-Schultz iteration will only be used to argue about the relationship between the condition number and the a priori pattern, and it will not be used to compute the a priori pattern.

easily shown that (see for example Theorem 3.1 in [1])

$$\|\mathcal{E}_k\|_2 \leq \left(\frac{\kappa^2 - 1}{\kappa^2 + 1} \right)^{2^k} \quad (22)$$

Similarly to (19), the Newton-Schultz iteration (20) tells us that the relatively good guess of the a priori pattern of \mathcal{A}^{-1} is given by the sum of powers of \mathcal{A} . This can be easily shown by starting from the initial guess (21), and by propagating the recursion (20). It should be remembered that from the (sparsity) pattern point of view, in which we only focus on the structure but not on the exact numerical values of the entries, the patterns produced by (19) and (20) can be made to be identical, and are mainly determined by the maximal powers of \mathcal{A} that are determined by the parameters z_1 and k . Furthermore, the bound (22) tells us that if the matrix \underline{A} is well-conditioned (it should be remembered that the condition numbers of \underline{A} and \mathcal{A} are equal), then the number k that produces a good approximation accuracy of the Newton-Schultz iteration is relatively small. Under the condition that \mathcal{A} is sparse multi-banded, small k or equivalently, small z_1 , produces sparse multi-banded \mathcal{C} . That is, the parameter z_1 in (19), that primarily determines the structure of the a priori pattern, should be small for well-conditioned \underline{A} .

Taking into account (6) and (19), we have that the pattern of the vectorized solution $\bar{\mathbf{x}}$ can be estimated by the pattern (or non-zero entries) of the vector $\bar{\mathbf{x}}_2$:

$$\bar{\mathbf{x}}_2 = \mathcal{C}\bar{\mathbf{p}} \quad (23)$$

For a large N , the vector $\bar{\mathbf{x}}_2$ cannot be computed directly from (23), simply because it is impossible to explicitly form the powers of \mathcal{A} . Instead, the vector $\bar{\mathbf{x}}_2$ should be expressed as the sum of $\mathcal{A}^l \bar{\mathbf{p}}$, $l = 1, 2, \dots, z_1$. Then, using the main properties of the Kronecker sum, these terms should be written as matrices (by reversing the vectorizing operation that is used to form (6)), and summed up together to form the matrix representation of $\bar{\mathbf{x}}_2$. For example, the term $\mathcal{A}\bar{\mathbf{p}}$, when transformed back to a matrix format, has the following form $\underline{A}\underline{P} + \underline{P}\underline{A}$. Similarly, the term $\bar{\mathbf{s}}_2 = \mathcal{A}^2 \bar{\mathbf{p}}$ can be written as

$$\bar{\mathbf{s}}_2 = (I \otimes \underline{A} + \underline{A} \otimes I) \bar{\mathbf{p}}_1, \quad \bar{\mathbf{p}}_1 = (I \otimes \underline{A} + \underline{A} \otimes I) \bar{\mathbf{p}} \quad (24)$$

and when transformed back to a matrix format \underline{S}_2 , the term $\bar{\mathbf{s}}_2$ has the following form:

$$\underline{S}_2 = \underline{A}\underline{P}_1 + \underline{P}_1\underline{A}, \quad \underline{P}_1 = \underline{A}\underline{P} + \underline{P}\underline{A} \quad (25)$$

The equation (25) implies that the entries $\mathcal{A}^l \bar{\mathbf{p}}$ can be computed recursively, involving Lyapunov-like operators⁵ of the form (25). That is, the a priori pattern, de-

⁵ This is not a Lyapunov equation because \underline{P} is known.

defined by (23), can be efficiently computed by performing operations on $(Nn) \times (Nn)$ sparse (multi) banded matrices. Our numerical experience shows that the complexity of computing a priori pattern for small values of z_1 , is negligible compared to the complexity of the methods for approximating \underline{X} , presented in Section 4.

Finally, it should be observed that the initial guess of the a priori pattern of \underline{X} , given by (19), can be used for other classes of sparse matrices \underline{A} and \underline{P} , such as non-symmetric banded matrices for example.

4 Methods for computing sparse approximation to \underline{X}

Using the insights from previous sections, in this section we develop two methods for computing (sparse) banded approximations to \underline{X} .

4.1 First method

For presentation clarity and brevity, in this section we will restrict our attention to banded a priori patterns of \underline{X} . All the methods can be extended to multi-banded or even more general a priori patterns, see also Section 3.2 and numerical experiments in Section 5.3.

Due to the fact that the a priori pattern is a banded matrix, the entries of $\bar{\mathbf{x}}$ corresponding to the entries of \underline{X} outside the *a priori bandwidth*, should be eliminated. Let \underline{X}_y be a y -banded, $Nn \times Nn$, binary matrix whose non-zero entries denote the a priori pattern of \underline{X} . Let us assume that $y \ll nN$, that is, the matrix \underline{X}_y is sparse and banded. Define the vector $\mathbf{x}_y = \text{vec}(\underline{X}_y)$. Let the vector $\tilde{\mathbf{x}}_y \in \mathbb{R}^{N_1}$ be defined by taking non-zero elements of \mathbf{x}_y and stacking them on top of each other. Given that the matrix \underline{X}_y is sparse and banded, we have that $N_1 \ll (Nn)^2$ (N_1 is in the order of Nn). Let the matrix $\tilde{\mathbf{A}}_1 \in \mathbb{R}^{(Nn)^2 \times N_1}$ be defined by eliminating the columns of \mathbf{A} corresponding to the zero entries of \mathbf{x}_y . Depending on the bandwidth y and the pattern and the bandwidth of \underline{A} , this elimination of columns of \mathbf{A} might induce zero rows of $\tilde{\mathbf{A}}_1$, that together with the corresponding elements of $\bar{\mathbf{p}}$ can be further eliminated. After these row eliminations of $\tilde{\mathbf{A}}_1$ and $\bar{\mathbf{p}}$, we obtain the matrix $\tilde{\mathbf{A}}_2 \in \mathbb{R}^{N_2 \times N_1}$ and the vector $\bar{\mathbf{p}}_2 \in \mathbb{R}^{N_2}$. We assume that $N_2 \geq N_1$ and that N_2 is in the order of Nn . The approximate solution to the Lyapunov equation can be found by solving the following least-squares problem:

$$\min_{\tilde{\mathbf{x}}} \|\bar{\mathbf{p}}_2 - \tilde{\mathbf{A}}_2 \tilde{\mathbf{x}}\|_2^2 \quad (26)$$

where $\tilde{\mathbf{x}} \in \mathbb{R}^{N_1}$. The solution of (26) can be found by solving the normal system of equations:

$$\hat{\mathbf{A}} \tilde{\mathbf{x}} = \hat{\mathbf{A}}_2^T \bar{\mathbf{p}}_2 \quad (27)$$

where $\hat{\mathbf{A}} = \tilde{\mathbf{A}}_2^T \tilde{\mathbf{A}}_2$ and $\hat{\mathbf{A}} \in \mathbb{R}^{N_1 \times N_1}$. The problem (26)-(27) can be efficiently solved using the Conjugate Gradient Least-Squares (CGLS) method [45, 46], without the need to explicitly form the normal equations. One iteration of the CGLS method takes about $2\text{nz}(\tilde{\mathbf{A}}_2) + 3N_1 + 2N_2$ flops, where $\text{nz}(\tilde{\mathbf{A}}_2)$ stands for the number of non-zero elements of $\tilde{\mathbf{A}}_2$ [46].

It is well-known that the convergence of the CG methods is fast for well-conditioned problems [45]. More precisely, the convergence of the CGLS method is determined by the factor⁶ $(\kappa_1 - 1)/(\kappa_1 + 1)$, where κ_1 is the condition number of $\hat{\mathbf{A}}$, for more details see Chapter 7 of [46]. Due to the fact that the matrix $\tilde{\mathbf{A}}_2$ is formed from the matrix \mathbf{A} and $\hat{\mathbf{A}} = \tilde{\mathbf{A}}_2^T \tilde{\mathbf{A}}_2$, it follows that the condition number of $\hat{\mathbf{A}}$ is related to the condition number of \mathbf{A} or to the condition number of \underline{A} . However, a theoretical study of this relationship is nontrivial and it is left for further research. Our numerical simulations indicate that if the matrix \mathbf{A} is well-conditioned, the matrix $\tilde{\mathbf{A}}_2$ inherits this numerical property. Furthermore, the matrix $\hat{\mathbf{A}}$ is also relatively well-conditioned (although the condition number of $\hat{\mathbf{A}}$ is usually larger than the condition number of \mathbf{A}). On the other hand, we showed that for a well-conditioned \underline{A} , the entries of \underline{X} decay quickly away from a banded pattern or from the main diagonal, and consequently, the bandwidth y of the a priori pattern \underline{X}_y can be chosen to be much smaller than Nn , without seriously compromising the accuracy. That is, for a well-conditioned \underline{A} , we have that N_1 and N_2 are naturally in the order of Nn . All these observations, together with the experience gathered by performing numerical simulations, indicate that for well-conditioned problems, the solution to (27) can be determined efficiently, with $O(N)$ computational and memory complexities, see also Remark 3. Theoretical analysis that can support our observations is left for a future research.

Remark 3 For a relatively ill-conditioned \underline{A} , the convergence rate of the CGLS method can be improved by employing the preconditioning techniques. To preserve the sparsity of the problem, techniques that employ sparse approximate inverse preconditioners can be used [47–49].

4.2 Second method

Here we will develop a second method for approximating \underline{X} . This method is based on the approximation of $\exp(t\underline{A})$ by a banded matrix. Using this approximation we approximate the integral representation (4). Such a solution is then used as an initial guess for the gradient projection method.

⁶ Factor closer to one implies slower convergence, and closer to zero means faster convergence.

In [32, 35] it has been shown that the matrix exponential of a symmetric banded matrix is an off-diagonally decaying matrix. A relatively non-conservative upper bound on the off-diagonal decay rate of the matrix exponential is derived in Theorem 4.2 in [35]. This decay bound depends on the extreme eigenvalues a and b of \underline{A} . However, the influence of the condition number on the decay rate of $\exp(t\underline{A})$, is still an open problem⁷. Due to the fact that $\exp(t\underline{A})$ is an off-diagonally decaying matrix, it can be approximated by a banded matrix using the Chebyshev series [32, 50], see also Remark 5. First, we transform the matrix function $\exp(t\underline{A})$ into a complex function $\exp(tz)$, where z is a complex number, belonging to a domain that contains the spectrum of \underline{A} , that is, $z \in [a, b]$. It is obvious that the eigenvalues of $t\underline{A}$ are in the interval $[ta, tb]$. The next step is to transform this interval into the interval $[-1, 1]$. This can be achieved by defining a new variable w as follows:

$$w = \frac{2tz - t(a+b)}{t(b-a)} \quad (28)$$

It can be easily seen that when $tz \in [ta, tb]$ then $w \in [-1, 1]$. From (28) we have:

$$tz = \frac{t}{2}((b-a)w + a + b) \quad (29)$$

Using (29), we have:

$$\exp(tz) = \exp\left(\frac{t}{2}((b-a)w + a + b)\right) \quad (30)$$

Let \underline{A}_1 be a matrix corresponding to the complex variable w . By substituting w with \underline{A}_1 and z by \underline{A} in (28), we obtain:

$$\underline{A}_1 = \frac{2}{b-a}\underline{A} - \frac{a+b}{b-a}I \quad (31)$$

The eigenvalues of \underline{A}_1 belong to the interval $[-1, 1]$. Similarly, from (29) we obtain:

$$t\underline{A} = \frac{t}{2}((b-a)\underline{A}_1 + (a+b)I) \quad (32)$$

By substituting $t\underline{A}$ in $\exp(t\underline{A})$ with (32), we define the function $f(\underline{A}_1)$:

$$f(\underline{A}_1) = \exp\left(\frac{t}{2}((b-a)\underline{A}_1 + (a+b)I)\right) \quad (33)$$

It is obvious that $f(\underline{A}_1) = \exp(t\underline{A})$. Consequently, the behavior of $f(\underline{A}_1)$ is identical to the behavior of $\exp(t\underline{A})$.

⁷ It should be expected that the decay rate is faster for well-conditioned matrices, because it very well known that the decay rate of matrix functions, such as \underline{A}^{-1} , is faster if the condition number of \underline{A} is smaller.

Consider the matrix \underline{A}_1 defined in (31). Taking into account that the spectrum of \underline{A}_1 belongs to the interval $[-1, 1]$, the truncated Chebyshev series expansion of the matrix exponential (33) is defined by [32, 51]:

$$f(\underline{A}_1) \approx \tilde{f}(\underline{A}_1), \\ \tilde{f}(\underline{A}_1) = \frac{c_1}{2}I + \sum_{k=2}^M c_k T_k(\underline{A}_1) = \sum_{k=1}^M {}' c_k T_k(\underline{A}_1) \quad (34)$$

where $c_k \in \mathbb{R}$ are the Chebyshev coefficients, $T_k(\underline{A}_1) \in \mathbb{R}^{N_n \times N_n}$ are the Chebyshev (matrix) polynomials of the first kind, and the symbol $\sum_{k=1}^M {}'$ means that the first term in the sum is halved [52]. The Chebyshev matrix polynomials are defined by [32, 51]:

$$T_1 = I, T_2 = \underline{A}_1, T_{k+1} = 2\underline{A}_1 T_k - T_{k-1}, k = 2, 3, \dots \quad (35)$$

The Chebyshev coefficients can be approximated by [32, 51] (see Remark 4):

$$c_k \approx \frac{2}{R} \sum_{j=1}^R f(\cos(\theta_j)) \cos((k-1)\theta_j) \quad (36)$$

where $\theta_j = \pi(j - \frac{1}{2})/R$ and R is a sufficiently large positive integer. The function $f(\cos(\theta_j))$ in (36) is defined as follows. First, by formally substituting in (33) the matrix \underline{A}_1 with w , and I with 1, we define:

$$f(w) = \exp\left(\frac{t}{2}((b-a)w + a + b)\right) \quad (37)$$

Then, substituting in (37) the argument w with $\cos(\theta_j)$ we obtain:

$$f(\cos(\theta_j)) = \exp(t\hat{q}_j), \hat{q}_j = \frac{(b-a)\cos(\theta_j) + a + b}{2}$$

Remark 4 An alternative method for computing the Chebyshev coefficients of the exponential function is based on the Bessel functions, see the equation (2.1) in [50]. \square

Remark 5 In the case of non-symmetric matrices \underline{A} , the Chebyshev approximation method cannot be directly used to approximate the matrix exponential. In the case of non-symmetric problems, the Faber polynomials need to be used, as demonstrated in [53].

The Chebyshev approximation error is defined by $\epsilon_M = \|f(\underline{A}_1) - \tilde{f}(\underline{A}_1)\|_2$. It can be easily shown that (see for

example, Section 4.1 in [32]):

$$\epsilon_M \leq \sum_{k=M+1}^{\infty} |c_k| \quad (38)$$

That is, the approximation error does not depend (explicitly) on the dimensions of \underline{A}_1 . On the other hand, in [50] it has been shown that:

$$\epsilon_M \leq \sum_{k=M+1}^{\infty} |c_k| \leq 2 \exp(tb) K_2(M, a, t) \quad (39)$$

where $K_2(M, a, t)$ is a constant depending on M, b and t , for more details see the equations (2.2) and (2.3) in [50]. By approximately computing the right-hand side of (38), or by computing (39), we can find the maximum order of the Chebyshev polynomials M for which the approximation error is below a predefined small number. However, if M is large, then the Chebyshev approximation $\tilde{f}(\underline{A}_1)$ is a dense matrix. Namely, from (35) we see that each Chebyshev polynomial can be expressed as the sum of powers of \underline{A}_1 . Due to the fact that \underline{A}_1 is m -banded, \underline{A}_1^k is km -banded, $k = 1, \dots, M$. That is, for large k , the matrix \underline{A}_1^k becomes fully populated. Given that N is large, it might be impossible to compute and to store \underline{A}_1^k . This implies that in order to ensure that the matrix $\tilde{f}(\underline{A}_1)$ is (sparse) banded and to guarantee that it can be computed and stored with $O(N)$ complexity, M needs to be kept small. If the off-diagonal decay rate of $\exp(t\underline{A})$ is fast, then our numerical results show that even for a small M , the approximation accuracy is relatively good. However, if the off-diagonal decay rate is not fast, selecting a relatively small M can seriously compromise the approximation accuracy. One of the ways to resolve this problem is to employ the *numerical dropping* technique [1, 32]. Namely we introduce the following modification in (35):

$$T_{k+1} = \mathcal{D}(2\underline{A}_1 T_k - T_{k-1}), \quad k = 2, 3, \dots \quad (40)$$

where the *bandwidth projection operator* $\mathcal{D}(Z)$, acting on an arbitrary matrix $Z = [z_{i,j}]$, is defined by:

$$\mathcal{D}(Z) = \begin{cases} z_{i,j}, & |i - j| \leq d/2 \\ 0, & |i - j| > d/2 \end{cases} \quad (41)$$

where d is the prescribed bandwidth. By setting to zero entries of $2\underline{A}_1 T_k - T_{k-1}$ that are outside the bandwidth d (that should be relatively small compared to Nn), the bandwidth projection operator ensures that each Chebyshev polynomial T_{k+1} remains a (sparse) banded matrix. In this way we can select a large M and ensure that the matrix $\tilde{f}(\underline{A}_1)$ can be computed with $O(N)$ memory and computational complexities, for more details see Section 4.3. The parameter d can be selected using the results

of Theorem 4.2 in [35], or using simple heuristics. Moreover, using the results of [35–37] it is possible to construct a projection operator in (40), that projects T_{k+1} onto a more optimal matrix pattern. Finally, the results of Section 3.2 can be extended to construct a more optimal projection operator.

We have observed that for small values of d and for relatively ill-conditioned \underline{A} , the modified Chebyshev recurrence (40) might diverge. To improve the accuracy of the Chebyshev approximation and to increase the stability of (40), instead of applying $\mathcal{D}(\cdot)$ in every iteration of (40), it can be applied in every second or third iteration. However, this strategy increases the computational complexity.

After we presented the method for approximating $\exp(t\underline{A})$, we turn our attention to the problem of approximating the integral in (4). From Theorem 4.3 in [54], we have that \underline{X} can be approximated by:

$$\underline{\tilde{X}} = - \sum_{j=-q}^q \psi \omega_j \exp(\psi t_j \underline{A}) \underline{P} \exp(\psi t_j \underline{A}) \quad (42)$$

where q is a positive integer, and

$$\psi = \frac{3}{2|b + \epsilon_1|}, \quad \omega_j = \left(q + q \exp\left(-2jq^{-1/2}\right) \right)^{-1/2} \\ t_j = \log \left(\exp\left(jq^{-1/2}\right) + \sqrt{1 + \exp\left(2jq^{-1/2}\right)} \right) \quad (43)$$

where $\epsilon_1 \ll |b|$ is a small number. In [54] it has been shown that the approximation error exponentially decreases with \sqrt{q} :

$$\|\underline{X} - \underline{\tilde{X}}\|_2 \leq K(\underline{A}) \|\underline{P}\|_2 \exp(-\sqrt{q}) \quad (44)$$

where the constant $K(\underline{A})$ depends on a and b . That is, as q approaches infinity, the approximation error approaches zero.

Consider the matrix $\exp(\psi t_j \underline{A})$ in (42). Let $\tilde{t}_j = \psi t_j$. For each \tilde{t}_j we can compute the Chebyshev approximation $\tilde{f}_j(\underline{A}_1)$ of the matrix exponential $\exp(\tilde{t}_j \underline{A})$. By substituting in (42) the matrix $\exp(\tilde{t}_j \underline{A})$ with $\tilde{f}_j(\underline{A}_1)$, we define the approximate solution of the Lyapunov equation:

$$\underline{\tilde{X}}_1 = - \sum_{j=-q}^q \psi \omega_j \tilde{f}_j(\underline{A}_1) \underline{P} \tilde{f}_j(\underline{A}_1) \quad (45)$$

Let us assume that each of the matrices $\tilde{f}_j(\underline{A}_1)$, $j = -q, \dots, q$, are computed using the Chebyshev approximation with the bandwidth projection operator (40). Consequently, the matrices $\tilde{f}_j(\underline{A}_1)$ are sparse, banded matrices with the bandwidth equal to d . Given that the

bandwidth of P is equal to l , we have that the total bandwidth of $\tilde{f}_j(\underline{A}_1) P \tilde{f}_j(\underline{A}_1)$ is $2d + l$. Given that $d \ll N$ and $l \ll N$, we have that $2d + l \ll N$, that is, the matrix $\tilde{f}_j(\underline{A}_1) P \tilde{f}_j(\underline{A}_1)$ is a sparse banded matrix. Furthermore, because the sum of matrices of equal bandwidths does not increase the bandwidth of the resulting sum, we have that \tilde{X}_1 is a sparse banded matrix with the bandwidth equal to $2d + l$. If $q \ll N$, then it is obvious that (45) can be computed with $O(N)$ computational and memory complexities, for more details see Section 4.3.

Next, we improve the accuracy of the approximate solution \tilde{X}_1 by using it as an initial guess of the decision variable of the constrained matrix least-squares problem [55]:

$$\min_{\underline{X}} \|\underline{P} - \underline{A}\underline{X} - \underline{X}\underline{A}^T\|_F^2 \quad (46)$$

$$\text{subject to } \underline{X} \in \mathcal{X}_{d_1} \quad (47)$$

where \mathcal{X}_{d_1} denotes the set of all banded matrices with the bandwidth of d_1 . The *gradient projection method* for solving (47) has the following form [40]:

$$\underline{X}_{k+1} = \mathcal{D}_1(\underline{X}_k - \delta_k \mathcal{G}_k), \quad k = 0, 1, 2, \dots \quad (48)$$

where $\mathcal{D}_1(\cdot)$ is the bandwidth projection operator defined in (41) for the bandwidth of d_1 , k is the iteration index, δ_k is the step size and \mathcal{G}_k is the gradient defined by [55]:

$$\mathcal{G}_k = -2\underline{A}^T \mathcal{R}_k - 2\mathcal{R}_k \underline{A} \quad (49)$$

$$\mathcal{R}_k = \underline{P} - \underline{A}\underline{X}_k - \underline{X}_k \underline{A}^T \quad (50)$$

To define the step size δ_k , we first define the following two quantities:

$$\begin{aligned} F_1(\underline{X}_k) &= \|\underline{P} - \underline{A}\underline{X}_k - \underline{X}_k \underline{A}^T\|_F^2, \\ \underline{X}_k(\delta) &= \mathcal{D}_1(\underline{X}_k - \delta \mathcal{G}_k) \end{aligned} \quad (51)$$

Keeping these definitions in mind, the step size is determined by the Armijo rule along the projection arc [40]:

$$\delta_k = \zeta^{h_k} \bar{\delta} \quad (52)$$

where h_k is equal to the first nonnegative integer h for which:

$$F_1(\underline{X}_k) - F_1(\underline{X}_k(\zeta^h \bar{\delta})) \geq \sigma \mathcal{G}_k^T(\underline{X}_k - \underline{X}_k(\zeta^h \bar{\delta})) \quad (53)$$

and $\sigma \in (0, 1)$, $\zeta \in (0, 1)$ and $\bar{\delta} > 0$. To summarize, the proposed method consists of the following two steps

(1) Compute the matrix \tilde{X}_1 defined in (45).

(2) Set $\underline{X}_0 = \tilde{X}_1$ and propagate the iteration (48) until convergence or the maximal number of iterations has been reached.

The convergence rate of the projected gradient method is well-studied in the literature, see for example [40]. Briefly speaking, the convergence rate is similar to the convergence rate of the unconstrained steepest descent method, and it depends on the condition number of \underline{A} . For well-conditioned problems the convergence is relatively fast. However, for ill-conditioned problems it can be slow. One of the ways to improve the convergence for ill-conditioned problems is to use scaled gradient projection methods, for more details see Chapter 2 of [40]. The optimization problem (46)-(47) can also be solved using other methods, for example using the matrix version of MINRES method with numerical droppings [45, 47, 56].

4.3 Complexity analysis

Let us analyze the complexity of the steps needed to compute the initial guess. The first computationally demanding step is to compute the extreme eigenvalues a and b of \underline{A} . This step can be performed efficiently using the ARPACK software or MATLAB functions `eigs` and `svds`, that are based on the Implicitly Restarted Arnoldi methods [57]. Generally speaking, one iteration of the methods implemented in ARPACK or in MATLAB, can be computed with $O(N)$ complexity, where the convergence rate depends on the spectral properties of \underline{A} . The storage requirement for computing the extreme eigenvalues is $O(N)$ [57]. Due to the fact that the initial guess is later on improved using the gradient projection method, we can tolerate some inaccuracies in computing extreme eigenvalues. That is, the tolerances for computing the extreme eigenvalues can be relaxed with the goal of achieving overall $O(N)$ computational complexity.

Let us now turn our attention to the Chebyshev approximation of the matrix exponential (34) and the integral approximation (45). In Appendix 6.1, we have shown that the computational complexity of these steps scales approximately linearly with N , M and q (assuming all other parameters are fixed), while on the other hand, it scales quadratically with d , where it is assumed that the bandwidth projection operator (40) is applied. For a relatively precise operations count see Appendix 6.1. Memory complexity scales with $O(N)$, assuming that $d \ll N$. Similarly it can be shown that the computational complexity of one iteration of the gradient projection method (48) scales linearly with N and d_1 , while the memory complexity scales with $O(N)$, assuming that $d_1 \ll N$.

5 Numerical Experiments

We demonstrate the effectiveness of the developed methods on three examples. Numerical simulations are performed in MATLAB on a laptop computer with 4 GB of

RAM, with processor specifications: Intel(R) Core(TM) i5-2410M CPU @ 2.30 GHz 2.30 GHz.

5.1 First example: 2D heat equation

We consider a model describing the temperature change of a thermally actuated deformable mirror used in extreme ultraviolet lithography [17, 58–60]. Heat equation constants, discretization grid and discretization steps are defined in Chapter 2 of [17]. The structure of this model is equivalent to a model obtained by the finite-difference discretization of the Laplace operator on a rectangular domain. The matrices \underline{A} and \underline{P} are block tri-diagonal, with the main block diagonals defined by:

$$A_{i,i} = \begin{bmatrix} a & e & 0 & 0 & 0 & 0 \\ e & a & e & 0 & 0 & 0 \\ 0 & e & a & e & 0 & 0 \\ 0 & 0 & e & a & e & 0 \\ 0 & 0 & 0 & e & a & e \\ 0 & 0 & 0 & 0 & e & a \end{bmatrix}, P_{i,i} = - \begin{bmatrix} 1 & 0.2 & 0.2 & 0.2 & 0.2 & 0.2 \\ 0.2 & 1 & 0.2 & 0.2 & 0.2 & 0.2 \\ 0.2 & 0.2 & 1 & 0.2 & 0.2 & 0.2 \\ 0.2 & 0.2 & 0.2 & 1 & 0.2 & 0.2 \\ 0.2 & 0.2 & 0.2 & 0.2 & 1 & 0.2 \\ 0.2 & 0.2 & 0.2 & 0.2 & 0.2 & 1 \end{bmatrix} \quad (54)$$

The off-diagonal blocks of \underline{A} are given by $A_{i,i-1} = A_{i,i+1} = eI_{6,6}$, where $a = -1.36$, $e = 0.34$ and $I_{6,6}$ is a 6×6 identity matrix. For the matrix \underline{P} we have: $P_{i,i-1} = P_{i,i+1} = [-0.1]$ (6×6 matrices whose entries are equal to -0.1). The total number of local subsystems N will be varied. The sparsity patterns of \underline{A} and \underline{P} , for $N = 10$, are shown in Fig.1.

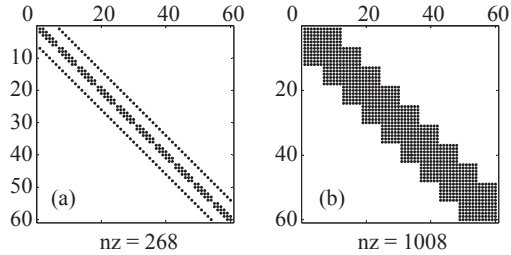


Fig. 1. Sparsity patterns: (a) \underline{A} . (b) \underline{P} , "nz" denotes the number of non-zero elements. Results are generated for $N = 10$.

We start with the Chebyshev approximation of the matrix exponential. For $t = 1$, we compute $\exp(t\underline{A})$ using the built-in MATLAB function `expm`(\cdot). The surface plot ("city plot") of this matrix is shown in Fig. 2.

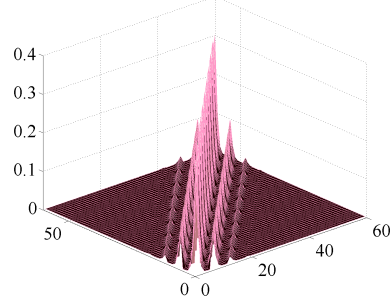


Fig. 2. Surface plot of $\exp(t\underline{A})$ for $t = 1$.

Next, for $t = 1$ and $N = 100$, we approximate $\exp(t\underline{A})$ using the Chebyshev method. For $M = 7$ in (34), the approximation error is $\epsilon_M = 4.4 \times 10^{-7}$, and the sparsity pattern of the approximate matrix exponential is shown in Fig. 3(a). These results confirm that $\exp(\underline{A})$ can be approximated by a sparse banded matrix with high accuracy [32, 42].

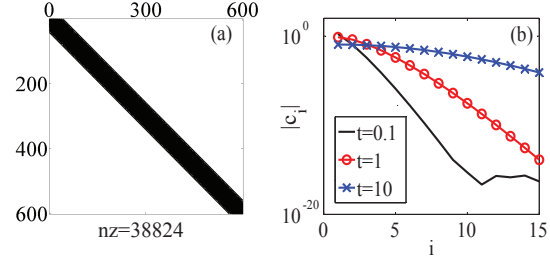


Fig. 3. (a) Sparsity pattern of the approximation of $\exp(\underline{A})$. (b) The dependence of the Chebyshev coefficients on t .

In Fig. 3(b), we illustrate the dependence of the Chebyshev coefficients on time t . It can be observed that as t increases, the Chebyshev coefficients decay more slowly. This numerically illustrates very-well known fact that for the fixed approximation order M , the accuracy of approximating $\exp(t\underline{A})$ using the Chebyshev approximation is better if t is smaller [32, 42]. This also implies that as t_j is increased in (42), to keep the accuracy of approximating $\exp(\psi t_j \underline{A})$ constant, we need to increase the order of the Chebyshev approximation M .

We now turn our attention to the problem of approximating \underline{X} . We will compare the approximate solution computed using the proposed methods with the "true" solution \underline{X}_T computed using the built-in MATLAB function `lyap`(\cdot). This solution is a dense matrix and its surface plot is shown in Fig. 4. The oscillatory behavior of entries of \underline{X}_T is due to the Kronecker sum structure of \underline{A} , see Section 3.2.

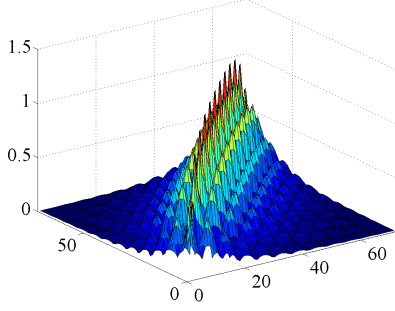


Fig. 4. The surface plot of the "true" solution \underline{X}_T computed using the function $\text{lyap}(\cdot)$. $N = 12$.

Our first goal is to numerically investigate the accuracy, and then the complexity of the proposed methods. We first investigate how the bandwidth of the approximate solutions affects the accuracy. We only show the results for the first method because the results for the second method are similar. The stopping criteria for the CGLS method is based on the following scalar [46]:

$$\eta = \frac{\|\tilde{\mathcal{A}}_2^T (\bar{\mathbf{p}}_2 - \tilde{\mathcal{A}}_2 \tilde{\mathbf{x}}^k)\|_2}{\|\tilde{\mathcal{A}}_2^T (\bar{\mathbf{p}}_2 - \tilde{\mathcal{A}}_2 \tilde{\mathbf{x}}^0)\|_2} \quad (55)$$

where $\tilde{\mathbf{x}}^k$ is the solution of (26)-(27) computed at the k -th iteration of the CGLS, and $\tilde{\mathbf{x}}^0$ is an initial guess that we chose as a zero vector. We stop the CGLS method when η is below 10^{-6} . Once the approximate solution has been computed with a prescribed tolerance, we quantify its (relative) accuracy by:

$$\varepsilon = \frac{\|\tilde{\underline{X}} - \underline{X}_T\|_2}{\|\underline{X}_T\|_2} \quad (56)$$

where $\tilde{\underline{X}}$ is an approximate solution. Figure 5(a) shows the dependence of the accuracy (56) on the bandwidth of the approximate solution. The results are generated for $N = 200$. As the bandwidth increases, the accuracy improves, as expected. It has been observed that the number of the iterations of the CGLS increases as the bandwidth increases (number of iterations to reach the stopping criteria defined by η). For example, for the bandwidth of 20 it is 45, and for the bandwidth of 300 it is 235. The condition number of \underline{A} is 39. Figure 5(b) shows the row 600 of the true and the approximate solution, and the error between them, computed for the bandwidth of 200.

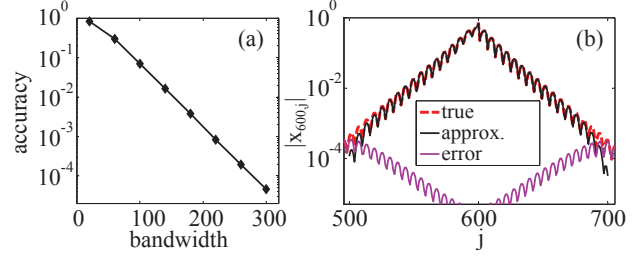


Fig. 5. (a) Dependence of the accuracy (56) on the bandwidth of the approximate solution. (b) Absolute value of the row 600 of the "true" and approximate solutions, and the error. Both (a) and (b) are generated for $N = 200$.

Next, we illustrate the dependence of the accuracy (56) on the parameter q in (45). We generate the results for $N = 250$, bandwidth of 140 in the iteration (40), and the Chebyshev order of $M = 20$. The approximation defined in (45) has a total bandwidth of 294. The results are shown in Fig. 6(a). As expected, the accuracy is improved by increasing q and it confirms the exponential dependence predicted by (44). In Fig. 6(b) we show the accuracy dependence on the number of iterations of the gradient projection method (48). The initial guess is computed on the basis of (45) for $q = 30$.

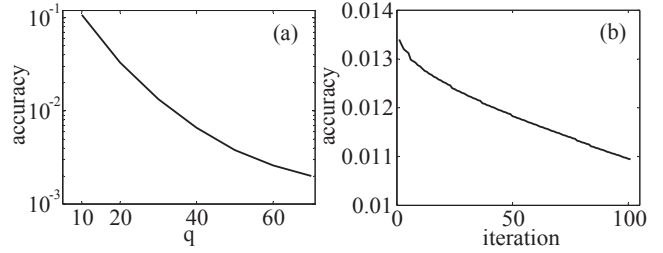


Fig. 6. (a) Dependence of the accuracy (56) on the parameter q in the equation (45). (b) Accuracy dependence on the number of iterations of the gradient projection method (48). Results are generated for $N = 250$ and bandwidth of 140.

Finally, we test the computational and memory complexities of the proposed methods. We vary N , and measure the time necessary to compute the approximate solutions. The results are generated for the bandwidth of 150. We also compare the complexity of the proposed methods with the complexity of the MATLAB function $\text{lyap}(\cdot)$. In order to compare the two methods, we chose their parameters such that their accuracy (56) is approximately equal and below 0.03 (for larger N it takes a lot of time or even it is impossible to compute the "true" solution, so these parameters are determined heuristically, such that the accuracy is guaranteed for $N \leq 600$). In the case of the first method, we stop the CGLS iteration when $\eta < 10^{-6}$. In the case of the second method, the parameters are: $q = 60$, $M = 20$ and the gradient projection iteration (48) is stopped after 50 iterations. Figure 7(a) shows the computational complexities of the

first and second methods, together with the computational complexity of the function $\text{lyap}(\cdot)$. Figure 7(b) shows the memory complexity.

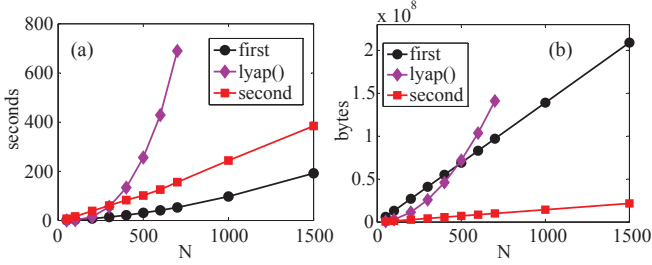


Fig. 7. Complexity of the first method, the second method and the MATLAB function $\text{lyap}(\cdot)$. (a) Computational complexity. (b) Memory complexity.

From 7(a) it can be observed that the first method has the lowest computational complexity. The second method has a higher computational complexity than the first method, however, its memory complexity is lower. Computational and memory complexities of the first and the second method scale approximately linearly with N . This enables us to compute the approximate solution for larger values of N . For figure clarity, we presented results for N up to 1500. However, we tested the second method for problems up to $N = 10^4$. The second method needs less than 50 minutes to compute the solution for $N = 10^4$. Due to the fact that we are not able to compute the "true" solution for such a large problem, we are not able to precisely quantify the accuracy of this approximate solution, but we expect that the accuracy is bounded by (44). Low memory complexity of the second method allows us to compute the solution even for much larger N (by extrapolating the results we estimate that for $N = 10^5$, the second method can compute the approximate solution in less than 8 hours). Finally, from Figure 7 we can observe the $O(N^3)$ computational and $O(N^2)$ memory complexities of the function $\text{lyap}(\cdot)$. This function has the highest computational and memory complexities.

5.2 Second example: randomly generated \underline{A}

Next, we illustrate the accuracy of the first method on a randomly generated model. Using the MATLAB function $\text{rand}(\cdot)$, we generate the matrices $A_{i,j}$, $j = i - 1, i, i + 1$ as a 6×6 random matrices. After constructing the block tri-diagonal matrix \underline{A} from these matrices, we define the following matrix $\underline{W} = \frac{1}{2}(\underline{A} + \underline{A}^T) + \nu I$, where the parameter ν has been chosen such that the matrix \underline{W} is asymptotically stable. The matrix \underline{P} is defined in the previous example. The surface plot of \underline{X}_T for the pair of the coefficient matrices $(\underline{W}, \underline{P})$ and for $N = 30$, is shown in Fig. 8(a).

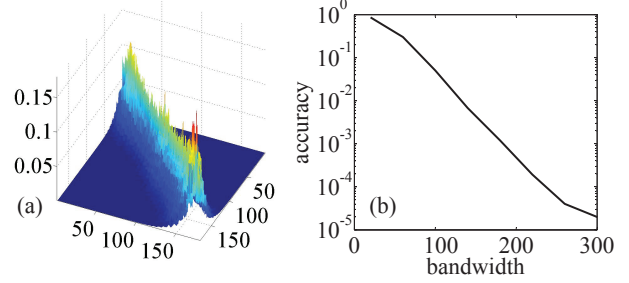


Fig. 8. Randomly generated model. (a) Surface plot of \underline{X}_T , $N = 30$. (b) Accuracy of the first method, $N = 200$.

From Fig. 8(a) we see that the off-diagonal decay rate of \underline{X} is fast, which directly follows from the fact that the matrix \underline{W} is well-conditioned (its condition number is approximately 52). Finally, in Fig. 8(b) we show the accuracy dependence on the bandwidth, that improves as the bandwidth increases. The second method gives similar results and for the sake of brevity is omitted.

5.3 Third example: 3D heat equation

We consider a 3D heat equation describing temperature change of a rectangular glass plate (diffusivity constant 3.4×10^{-7}) used in optical systems [17]. The heat equation is discretized using the finite difference method with the spatial discretization step of 0.001 (all the units are in the SI system). The dimension of the grid in the z direction is fixed to 6, whereas in the $x - y$ plane(s) it is defined by an $N_1 \times N_1$, where N_1 is the parameter that is varied. In total the grid has $6 \times N_1^2$ points. Such a discretization grid can be seen as an interconnection of $N = N_1^2$ subsystems $S_{i,j}$, where each subsystem's state consists of the temperatures in the z -direction: $\{T_{i,j,1}, T_{i,j,2}, \dots, T_{i,j,6}\}$. That is, the local order of each $S_{i,j}$ is $n = 6$, for more details see Chapter 2 of [17].

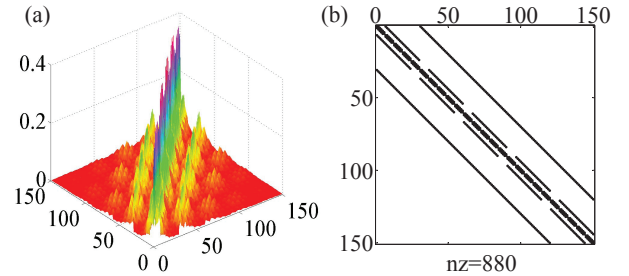


Fig. 9. Discretized 3D heat equation. (a) Surface plot of \underline{X}_T . (b) Sparsity pattern of \underline{A} . $N = 25$.

The matrix \underline{P} is equal to the matrix used in the first example. The surface plot of \underline{X}_T and sparsity pattern of \underline{A} are shown in Fig. 9(a) and 9(b), respectively, for $N = 25$, ($N_1 = 5$). Similarly to the first example, from

Fig. 9(a) we see that \underline{X}_T shows an off-diagonally decaying, oscillatory behavior.

We quantify the accuracy (56) of the first method (results obtained by the second method are similar). For the CGLS we use the tolerance $\eta = 10^{-6}$. We construct the model for $N_1 = 30$ which gives in total $N = 900$ local subsystems ($\underline{A} \in \mathbb{R}^{5400 \times 5400}$). The condition number of \underline{A} is 72 and it increases as N is increased. Figure 10(a) shows the accuracy for an a priori pattern equal to a banded matrix. As expected, the accuracy improves as the bandwidth increases. However, the improvement is slower compared to the first example, see Fig. 5(a). This is because \underline{X}_T for the third example, has more dominant off-diagonal peaks than in the first example, as it can be observed in Fig. 9(a). Similarly to the first example, we noticed that the number of CGLS iterations increases (iterations to reach the prescribed tolerance) as the bandwidth is increased. Figure 10(b) shows the accuracy for an a priori pattern equal to a multi-banded matrix computed using the methodology proposed in Section 3.2. Such an a priori pattern is shown in Fig. 10(d) for $z_1 = 8$. Finally, Fig. 10(c) shows the percentage of the non-zero elements of the approximate solution for the two types of patterns. From Figures 10(a)-(c) we conclude that multi-banded a priori pattern achieves better accuracy with a smaller number of non-zero elements compared to the purely banded a priori pattern.

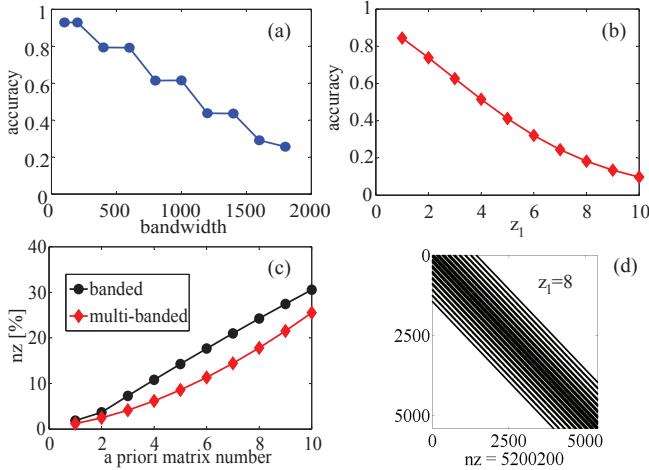


Fig. 10. Discretized 3D heat equation. (a) Accuracy (56) as a function of bandwidth for strictly banded a priori pattern. (b) Accuracy for the a-priori pattern computed using the method of Section 3.2 (multi-banded pattern shown in (d) for $z_1 = 9$). (c) Comparison of the percentage of the non-zero elements (denoted by "nz") of a-priori sparsity patterns used to generate plots (a) and (b). (d) Plot of the multi-banded a-priori sparsity pattern, whose performance is shown in (b), the pattern is generated using the method in Section 3.2 for $z_1 = 8$.

6 Conclusions

We considered the problem of computing a banded, approximate solution of the Lyapunov equation (1), in which the coefficient matrices are banded, symmetric matrices. We analyzed how the condition number of the coefficient matrix \underline{A} influences the decay rate of the solution \underline{X} . We showed that the decay rate is faster if the condition number of \underline{A} is smaller. We proposed two computationally efficient methods for approximating the solution by a banded matrix. Our results indicate that for a well-conditioned, sparse banded matrix \underline{A} , it is possible to compute a sparse, banded approximate solution of the Lyapunov equation with $O(N)$ complexity. In the future work, the proposed methods will be used to develop computationally efficient methods for approximating the solution of the Riccati equation by a sparse matrix.

Appendix

6.1 Computational complexity analysis of the second method

Due to the fact that we deal with (sparse) banded matrices, to estimate the computational complexity, we will not count multiplications and additions/subtractions of an arbitrary number with zero, because depending on the implementation (such as MATLAB sparse matrix toolbox), such operations do not require computing power. Let us assume that we have two banded, $Nn \times Nn$ matrices X_1 and X_2 with bandwidths d_1 and d_2 , respectively. The matrix resulting from their multiplication $X_3 = X_1 X_2$ is a banded matrix with the bandwidth of $d_3 = d_1 + d_2$. The rows of X_3 : $d_3/2 + 2, \dots, Nn - d_3/2 - 1$ have $d_1 + d_2 + 1$ non-zero entries, and remaining rows have smaller number of entries. For simplicity, we will assume that every row of X_3 has $d_1 + d_2 + 1$ non-zero entries. To compute every entry on such a row, it takes maximum $d_m + 1$ multiplications and d_m additions, where $d_m = \min(d_1, d_2)$. So in total, the number of operations that are necessary to compute X_3 is smaller than $Nn(d_1 + d_2 + 1)(2d_m + 1)$ operations. Under the same simplifications, we conclude that the number of operations necessary to compute $X_4 = X_1 \pm X_2$ is smaller than $Nn(d_m + 1)$, and number of operations necessary to multiply a scalar c and the matrix X_1 is smaller or equal to $Nn(d_1 + 1)$.

Let us now turn our attention to the Chebyshev approximation of the matrix exponential. The complexity of computing \underline{A}_1 using (31) is much smaller than the complexity of subsequent operations and for simplicity it will be ignored. Consider the iteration (40) in which T_k has a bandwidth of d (produced by applying the operator $\mathcal{D}(\cdot)$ in the iteration $k - 1$) and the matrix \underline{A} has a bandwidth of m , $d > m$. It takes less than $Nn(m + 1) + Nn(d + m + 1)(2m + 1)$ operations to compute $2\underline{A}_1 T_k$ (multiplication of \underline{A}_1 by a scalar and

multiplication of two banded matrices) plus additional $Nn(d+1)$ operations to compute $2\mathbf{A}_1T_k - T_{k-1}$. That is, the number of operations to compute one iteration (40) is less than $o_1 = Nn(m+d+2+(d+m+1)(2m+1))$. For simplicity, we will ignore the operation count of the operator $\mathcal{D}(\cdot)$, because it depends on the implementation details. Due to the fact that we need $M-2$ iterations to compute all the matrices T_k (we ignore complexity of computing T_1 and T_2), and because we need to multiply them with a scalar and add them together, the total number of operations to compute the approximation (34), is smaller than $o_2 = (M-2)o_1 + 2NnM(d+1)$. The resulting matrix has the bandwidth of d . Let us now focus on (45). Assuming that the bandwidth l of \underline{P} is smaller than d , and using similar reasoning it can be concluded that the number of iterations is smaller than

$$o_3 = (2q+1)(o_2 + Nn(d+l+1)(2l+1) + Nn(2d+l+1)(2d+1) + Nn(d+1))$$

The important conclusion from the above analysis is that the complexity approximately scales linearly with Nn , M and q , while on the other hand it scales quadratically with d .

Similar analysis can be performed for the gradient projection method. However, it is hard to give a relatively precise estimate, because the number of step-size evaluations (53) in each iteration, cannot be predicted a priori. However, assuming that the number of step-size evaluations is very small, and using analysis similar to the Chebyshev approximation, it can be concluded that the computational complexity of every step of the gradient projection method scales linearly with Nn and d_1 (there is no need to perform multiplications of two banded matrices of the bandwidths of d_1).

References

- [1] A. Haber and M. Verhaegen. Sparse Approximate Inverses of Gramians and Impulse Response Matrices of Large-scale Interconnected Systems. *arXiv preprint arXiv:1405.2580v1 [cs.SY]*, 2014.
- [2] S. K. Pakazad, A. Hansson, M. S. Andersen, and A. Rantzer. Distributed Robustness Analysis of Interconnected Uncertain Systems Using Chordal Decomposition. *arXiv preprint arXiv:1402.2066*, 2014.
- [3] R. D'Andrea and G. Dullerud. Distributed Control Design for Spatially Interconnected Systems. *IEEE Transactions on Automatic Control*, 48:1478–1495, 2003.
- [4] M. R. Jovanović and B. Bamieh. On the Ill-posedness of Certain Vehicular Platoon Control Problems. *IEEE Trans. Automat. Control*, 50(9):1307–1321, September 2005.
- [5] B. Bamieh, M. R. Jovanović, P. Mitra, and S. Patterson. Coherence in Large-Scale networks: Dimension Dependent Limitations of Local Feedback. *IEEE Trans. Automat. Control*, 57(9):2235–2249, September 2012.
- [6] B. Bamieh, O. Paganini, and M. A. Dahleh. Distributed Control of Spatially Invariant Systems. *IEEE Transactions on Automatic Control*, 47:1091–1118, 2002.
- [7] U. A. Khan and J. M. F. Moura. Distributing the Kalman Filter for Large-Scale Systems. *Signal Processing, IEEE Transactions on*, 56(10):4919–4935, 2008.
- [8] D. Gorinevsky, S. Boyd, and G. Stein. Design of Low-Bandwidth Spatially Distributed Feedback. *Automatic Control, IEEE Transactions on*, 53(1):257–272, 2008.
- [9] N. Motee and A. Jadbabaie. Optimal control of spatially distributed systems. *IEEE Transactions on Automatic Control*, 53:1616–1629, 2008.
- [10] N. Motee and Q. Sun. Measuring Sparsity in Spatially Interconnected Systems. In *Decision and Control (CDC), 2013 IEEE 52nd Annual Conference on*, pages 1520–1525. IEEE, 2013.
- [11] M. Siami and N. Motee. Graph-Theoretic Bounds on Disturbance Propagation in Interconnected Linear Dynamical Networks. *arXiv preprint arXiv:1403.1494*, 2014.
- [12] N. Matni. Communication Delay Co-Design in \mathcal{H}_2 Distributed Control Using Atomic Norm Minimization. *arXiv preprint arXiv:1404.4911*, 2014.
- [13] M.S. Andersen, S.K. Pakazad, A. Hansson, and A. Rantzer. Robust Stability Analysis of Sparsely Interconnected Uncertain Systems. *Automatic Control, IEEE Transactions on*, 59(8):2151–2156, Aug 2014.
- [14] T. Zhou. Coordinated One-Step Optimal Distributed State Prediction for a Networked Dynamical System. *Automatic Control, IEEE Transactions on*, 58(11):2756–2771, Nov 2013.
- [15] T. Zhou. On the Controllability and Observability of Networked Dynamic Systems. *Automatica*, 52:63–75, 2015.
- [16] P. Benner. Solving Large-Scale Control Problems. *Control Systems, IEEE*, 24(1):44 – 59, 2004.
- [17] A. Haber. *Estimation and Control of Large-Scale Systems With an Application to Adaptive Optics for EUV Lithography*. PhD thesis, Delft University of Technology, Delft, The Netherlands, 2014.
- [18] J. Sun and A. E. Motter. Controllability Transition and Nonlocality in Network Control. *Phys. Rev. Lett.*, 110:208701, May 2013.
- [19] Z. Gajic and M. T. J. Qureshi. *Lyapunov Matrix Equation in System Stability and Control*. Courier Corporation, 2008.
- [20] V. L. Mehrmann. *The Autonomous Linear Quadratic Control Problem: Theory and Numerical Solution*, volume 163. Springer, 1991.
- [21] V. Simoncini. Computational Methods for Linear Matrix Equations. *Technical Report. Dipartimento di Matematica. Università di Bologna.*, 2013.
- [22] D. A. Bini, B. Iannazzo, and B. Meini. *Numerical Solution of Algebraic Riccati Equations*, volume 9. SIAM, 2012.
- [23] F. Feitzinger, T. Hylla, and E. W. Sachs. Inexact Kleinman-Newton method for Riccati equations. *SIAM Journal on Matrix Analysis and Applications*, 31(2):272–288, 2009.
- [24] X. Wang, W.-W. Li, and L. Dai. On Inexact Newton Methods Based on Doubling Iteration Scheme for Symmetric Algebraic Riccati Equations. *Journal of Computational and Applied Mathematics*, 260:364–374, 2014.
- [25] P. Benner, J. R. Li, and T. Penzl. Numerical Solution of Large-Scale Lyapunov Equations, Riccati Equations, and Linear-Quadratic Optimal Control Problems. *Numerical Linear Algebra Appl*, 15(9):755–777, 2008.
- [26] D. A. Bini, B. Iannazzo, and F. Poloni. A Fast Newton's Method for a Nonsymmetric Algebraic Riccati Equation. *SIAM Journal on Matrix Analysis and Applications*, 30(1):276–290, 2008.

- [27] P. Benner and J. Saak. Numerical Solution of Large and Sparse Continuous Time Algebraic Matrix Riccati and Lyapunov Equations: A State of the Art Survey. Technical report, 2013.
- [28] P. Benner and H. Faßbender. On the Numerical Solution of Large-Scale Sparse Discrete-Time Riccati Equations. *Adv. Comput. Math.*, 35(2-4):119–147, 2011.
- [29] F. Lin, M. Fardad, and M. R. Jovanović. Design of Optimal Sparse Feedback Gains Via the Alternating Direction Method of Multipliers. *IEEE Trans. Automat. Control*, 58(9):2426–2431, September 2013.
- [30] F. Lin, M. Fardad, and M. R. Jovanović. Augmented Lagrangian Approach to Design of Structured Optimal State Feedback Gains. *IEEE Trans. Automat. Control*, 56(12):2923–2929, December 2011.
- [31] S. Schuler, P. Li, J. Lam, and F. Allgöwer. Design of Structured Dynamic Output-Feedback Controllers for Interconnected Systems. *International Journal of Control*, 84(12):2081–2091, 2011.
- [32] M. Benzi and N. Razouk. Decay Bounds and $O(n)$ Algorithms for Approximating Functions of Sparse Matrices. *Electronic Transactions on Numerical Analysis*, pages 16–39, 2007.
- [33] A. Haber and M. Verhaegen. Subspace Identification of Large-Scale Interconnected Systems. *Automatic Control, IEEE Transactions on*, 59(10):2754–2759, 2014.
- [34] S. Demko, W. F. Moss, and P. W. Smith. Decay Rates for Inverses of Band Matrices. *Mathematics of Computation*, 43(168):491–499, 1984.
- [35] M. Benzi and V. Simoncini. Decay Bounds for Functions of Matrices with Banded or Kronecker Structure. *arXiv preprint arXiv:1501.07376*, 2015.
- [36] C. Canuto, V. Simoncini, and M. Verani. On the Decay of the Inverse of Matrices that are Sum of Kronecker Products. *Linear Algebra and its Applications*, 452:21–39, 2014.
- [37] V. Simoncini. The Lyapunov Matrix Equation. Matrix Analysis from a Computational Perspective. *arXiv preprint arXiv:1501.07564*, 2015.
- [38] M. Verhaegen and V. Verdult. *Filtering and System Identification: A Least Squares Approach*. Cambridge University Press, 2007.
- [39] A. J. Laub. *Matrix Analysis for Scientists and Engineers*. Siam, 2005.
- [40] D. P. Bertsekas. *Nonlinear programming*. Athena Scientific, 1999.
- [41] M. Shao. On the Finite Section Method for Computing Exponentials of Doubly-Infinite Skew-Hermitian Matrices. *Linear Algebra and its Applications*, 451:65–96, 2014.
- [42] M. Benzi and G. H. Golub. Bounds for the Entries of Matrix Functions with Applications to Preconditioning. *BIT Numerical Mathematics*, 39(3):417–438, 1999.
- [43] T. Huckle. Approximate Sparsity Patterns for the Inverse of a Matrix and Preconditioning. *Applied numerical mathematics*, (30):291–303, 1999.
- [44] V. Pan and R. Schreiber. An Improved Newton Iteration for the Generalized Inverse of a Matrix with Applications. *SIAM Journal on Scientific and Statistical Computing*, 12(5):1109–1130, 1991.
- [45] Y. Saad. *Iterative Methods for Sparse Linear Systems*. Society for Industrial and Applied Mathematics, 2003.
- [46] A. Björck. *Numerical methods for least squares problems*. Siam, 1996.
- [47] E. Chow and Y. Saad. Approximate Inverse Preconditioners via Sparse-Sparse Iterations. *SIAM Journal on Scientific Computing*, 19(3):995–1023, 1998.
- [48] E. Chow. A Priori Sparsity Patterns for Parallel Sparse Approximate Inverse Preconditioners. *SIAM Journal on Scientific Computing*, 21(5):1804–1822, 2000.
- [49] M. Benzi and M. Tuma. A Sparse Approximate Inverse Preconditioner for Nonsymmetric Linear Systems. *SIAM Journal on Scientific Computing*, 19(3):968–994, 1998.
- [50] L. Bergamaschi and M. Vianello. Efficient Computation of the Exponential Operator for Large, Sparse, Symmetric Matrices. *Numerical linear algebra with applications*, 7(1):27–45, 2000.
- [51] J. C. Mason and D. C. Handscomb. *Chebyshev Polynomials*. Chapman & Hall/CRC, 2003.
- [52] R. J. Mathar. Chebyshev Series Expansion of Inverse Polynomials. *J. Comput. Appl. Math.*, 196(2):596–607, November 2006.
- [53] L. Bergamaschi, M. Caliari, and M. Vianello. Efficient Approximation of the Exponential Operator for Discrete 2D Advection–Diffusion Problems. *Numerical linear algebra with applications*, 10(3):271–289, 2003.
- [54] L. Grasedyck, W. Hackbusch, and B.N. Khoromskij. Solution of Large Scale Algebraic Matrix Riccati Equations by Use of Hierarchical Matrices. *Computing*, 70(2):121–165, 2003.
- [55] Z.-Y. Li and Y. Wang. Weighted Steepest Descent Method for Solving Matrix Equations. *International Journal of Computer Mathematics*, 89(8):1017–1038, 2012.
- [56] P. Benner, J. Saak, M. Stoll, and H. K. Weichelt. Efficient Solution of Large-Scale Saddle Point Systems Arising in Riccati-based Boundary Feedback Stabilization of Incompressible Stokes Flow. *SIAM Journal on Scientific Computing*, 35(5):S150–S170, 2013.
- [57] R. B. Lehoucq, D. C. Sorensen, and C. Yang. *ARPACK Users’ Guide: Solution of Large-scale Eigenvalue Problems with Implicitly Restarted Arnoldi Methods*, volume 6. SIAM, 1998.
- [58] A. Haber, A. Polo, S. K. Ravensbergen, H. P. Urbach, and M. Verhaegen. Identification of a Dynamical Model of a Thermally Actuated Deformable Mirror. *Opt. Lett.*, 38(16):3061–3064, Aug 2013.
- [59] A. Haber, A. Polo, I. Maj, S.F. Pereira, H.P. Urbach, and M. Verhaegen. Predictive Control of Thermally Induced Wavefront Aberrations. *Optics express*, 21(18):21530–21541, 2013.
- [60] A. Haber, A. Polo, C. S. Smith, S. F. Pereira, P. Urbach, and M. Verhaegen. Iterative Learning Control of a Membrane Deformable Mirror for Optimal Wavefront Correction. *Appl. Opt.*, 52(11):2363–2373, Apr 2013.